REMARKS

This application has been carefully considered in connection with the Office Action dated September 18, 2008. Reconsideration and allowance are respectfully requested in view of the following.

Summary of Rejections

Claims 1-33 were pending at the time of the Office Action.

Claims 1-33 were rejected under 35 USC § 103.

Claims 1, 9, and 28 were also rejected under 35 USC § 102.

Summary of Response

No claims are currently amended.

Claims 1, 3, 9-11, 23, 28, and 31 were previously presented.

Claims 2, 4-8, 12-22, 24-27, 29-30 and 32-33 remain as originally submitted.

Remarks and Arguments are provided below.

Summary of Claims Pending

Claims 1-33 are currently pending following this response.

Response to Rejections

Upton, Beck, O'Donnell and Bhatia, alone or in combination, do not disclose, teach or suggest a client application on a first operating system obtaining a token containing user credentials encoded as a platform and application independent string data type and providing the token to a server application on a second operating system to use services of the server application, as claimed. Making the token a string data type makes it platform and application independent because the token has no header and, therefore, no application-specific header configuration. This feature of the claimed token eliminates the need to convert security information from the format of one platform to the format of another.

The system of the pending application includes a security application program interface, an authentication authority, a store maintaining data, an application program interface, and a server application. The security application program interface and application program interface are coupled to a client application on a first operating system. The security application program interface provides a security credential. The authentication authority receives the security credential from the security application program interface and, if the security credential is valid, the authentication authority generates a token and communicates the token to the security application program interface. The store maintaining the data, which is in communication with the authentication authority, validates the security credential. The application program interface is coupled to the client application and can communicate regarding the token. The server application is on a second operating system and receives the token from the application program interface and communicates with the authentication authority to validate the token to enable the client application to use services of the server application.

This approach to providing application-to-application enterprise security permits the communication of security information in a heterogeneous computing environment using a token containing user credentials encoded as a platform and application independent string data type.

With regard to the art rejections, the Office Action has rejected the pending claims citing Upton in view of Beck and further in view of O'Donnell. The Office Action has also rejected the pending claims citing Upton in view of Beck and further in view of Bhatia. Upton relates to systems and methods for integration adapter security. Beck relates to methods and apparatus for providing anonymity to end users in web transactions. O'Donnell relates to methods and apparatus for securely granting access rights to unattended software. Bhatia relates to systems and methods for sign-on to Web-based applications. Upton, Beck, O'Donnell and Bhatia do not disclose, teach or suggest using a token containing user credentials encoded as a platform and application independent string data type, or using a security application program interface and an application program interface coupled to a client application on a first operating system, and a server application on a second operating system to receive the token from the application program interface and communicate with an authentication authority to validate the token, as claimed. Notably, the tokens found in the pending disclosure permit the authentication of users in a heterogeneous computing environment. In contrast, the computing environments in Upton, Beck, O'Donnell and Bhatia require a homogeneous computing environment.

These distinctions, as well as others, will be discussed in greater detail in the analysis of the present claims that follows.

Response to Rejections under Section 103

Claim 1:

Claim 1 was rejected under 35 USC § 103(a) as being unpatentable over Upton, U.S. Patent Application Publication No. 2003/0097574 (hereinafter, "Upton") in view of O'Donnell et al., U.S. Patent Application Publication No. 2004/0117615 (hereinafter, "O'Donnell").

Claim 1 was also rejected under 35 USC §103(a) as being unpatentable over Upton in view of Bhatia et al., U.S. Patent Application Publication No. 2003/0200465 (hereinafter, "Bhatia").

I. Upton, Beck, O'Donnell and Bhatia do not disclose, teach or suggest wherein the token contains user credentials encoded as a platform and application independent string data type.

Claim 1 (as previously presented) recites, in part, "wherein the token contains user credentials encoded as a platform and application independent string data type..."

With respect to claim 1, the Office Action states, in pertinent part:

> ***Regarding claim 1,*** Upton discloses a system to provide application-to-application enterprise security, the system comprising:
>
> ...an authentication authority (Par 0063-0065, 0104, 0128, 0145,; SAS, or JAAS, or authentication/ authorization SPI) receiving the security credential (Par 0061-0069; credentials; security-pinciple; ra.xml file) from the security

> application program interface, the authentication authority further operable to communicate the token to the security application program interface where the security credential is valid, wherein the token contains user credentials encoded as a platform and application independent string data type (fig 4; Par 0063-0069, 0104, 0114, 0130, 0150; service provider interface/ SPI; checking public/ password type, or generic token type credentials, or security-principal map element).

Office Action dated September 18, 2008, Page 3.

Contrary to the assertions made in the Office Action, Upton does not disclose, teach or suggest "wherein the token contains user credentials encoded as a platform and application independent string data type," as recited in (previously presented) claim 1. Specifically, the Office Action relied on the following disclosure in Upton to read on this element of claim 1:

> [0150] A Principal→Credential Mapping (JAAS Login-Module) SPI can be based on the JAAS Login Module, and can be used to map principal identity when cross security domain policy or technology boundaries. The responsibilities of the Principal Mapping SPI is based on the Subject provided, and can be used to add public credentials with appropriate information to subject, such as password credentials for username/password, and <u>generic credential for token-type credentials</u>. (Underlining added for emphasis.)

The disclosure in Upton of "generic credential for token-type credentials" is not the token recited in claim 1. For example, the American Heritage® Dictionary of the English Language, Fourth Edition, defines the term "generic" as "Relating to or descriptive of an entire group or class; general." Clearly, a generic credential (for token-type credentials) relating to or descriptive of an entire group or class does not teach or suggest "wherein the token contains user credentials encoded as a platform and application independent string data type," as claimed.

Furthermore, a text search of Upton for the string "generic" reveals that the word is used only twice: once in paragraph 0150 cited above and once in paragraph 0065. Upton, paragraph 0065 recites that "the resource adapters can support: password credentials and generic credentials." It is clear from the context that "generic credential" in no way refers to "platform and application independent" credentials, but is merely contrasting "generic credential" with "password credential". Thus, "generic credentials" are merely non-password credentials.

Also, the system of Upton is based on mapping credentials from a user to a resource. For example, paragraph 0087 of Upton recites:

> The "EIS Sign-on" section of the J2EE Connector Specification, Version 1.0 Final Release identifies a number of possible options for defining a Resource Principal on whose behalf the sign-on is being performed. Previous implementations implemented the Security Principal Map option identified in the specification. Under this option, a resource principal is determined by mapping from the identity of the initiating/caller principal for the invoking component. The resultant resource principal does not inherit the identity or security attributes of the principal that it is mapped from, but instead gets its identity and security attributes (password) based upon the defined mapping.

However, this method is contrary to the elements of claim 1 which do not require a mapping of a credential from one application to another. To better illustrate this and better understand the contrast between Upton and claim 1, paragraphs 0027 and 0028 of the pending application are provided below which state:

> **[0027]** The present system allows tokens to be passed among disparate applications so that security information can automatically be included with each call from one application to another. This eliminates the need for conversion of security information in message headers between the data format of the applications. It also eliminates the need for an application to be authenticated and authorized every time it sends a message to another application. In contrast with services where a security context remains present on a server, in embodiments of the invention there is no permanent context or session. Instead, a context is created with every invocation from one application to another.
>
> **[0028]** Rather than security information being converted from the format of one platform to the format of another, security information is passed between applications in the form of a token with a string data type. Since a string is a primitive data type, it can be recognized by a large number of applications and interfaces, meaning it can be sent over multiple services such as J2EE, CORBA, and IBM's MQSeries. Making the token a string makes it platform and technology independent because the token has no header and therefore no application-specific header configuration.

Mapping credential information as Upton teaches is equivalent to conversion of security information and is different from the token of claim 1 "wherein the token contains user credentials encoded as a platform and application independent string data type."

Additionally, nothing in this passage discloses that the "token contains user credentials encoded as a platform and application independent <u>string data type</u>" (emphasis added). In fact, the Office Action is silent with regard to the claimed "string data type." Upton does not disclose the data type for the credentials and does not suggest that the data type may be a "string data type."

The Office Action also relied on the following disclosure in Upton to read on the same element of claim 1:

> [0104] For many systems, it may be desirable to include support for the Java 2 Enterprise Edition (J2EE) specification and interoperability therewith. These J2EE specification features include the Common Secure Interoperability (CSI) protocol, user identity tokens, the Stateless Authentication Service (SAS) protocol, support for propagation of security credentials across machine, cluster, and/or domain boundaries, control of propagation of identity based on policy, enhanced support for virtual host/sites, the ability to generate a user identity scoped to domain, and host/site specific security policies. (Underlining added for emphasis.)

As shown in the above-cited paragraph, Upton teaches the use of "user identity tokens" and does not teach or suggest "wherein the token contains user credentials encoded as a platform and application independent string data type," as recited in claim 1.

Beck teaches the use of a temporary user ID token. However, Beck discloses in paragraph [0020], lines 1-7, "As described above, each time the end-user's browser 104 makes a request to any web site with which ISP 102 has an established relationship, that request is modified by the inclusion of an HTTP header containing a temporary user ID token. That token enables the end-user to maintain his anonymity to such Web site by using the trustworthy ISP as his agent in the transaction." (Underlining added for emphasis.)

Beck also discloses in paragraph [0021], lines 1-9, "In a second embodiment of the invention, HTTP cookies rather than HTTP headers are used as the mechanism to transport a temporary user ID token (and possibly other information) to third-party applications running on a Web server. ...A cookie is introduced to the client by including a Set-Cookie header as part of an HTTP response."

Beck further discloses in paragraph [0022], lines 18-23 and 29-32, "A service module 206 within intermediary 108 then modifies the response to include a Set-Cookie header. That header includes a temporary random or pseudo-random user ID token that is generated by the service module 206 and is assigned to the end-user until the cookie's expiration date. ...Intermediary 108 then forwards the modified response containing the Set-Cookie header with its associated temporary user ID token and additional information to the end-user's Web browser 104." (Underlining added for emphasis.)

As shown in the above-cited paragraphs, Beck teaches the use of associating a header with its temporary user ID token. In contrast to Beck's teachings of a temporary user ID token, the specification of the pending application states in paragraph [0028], lines 1-3 and 5-7, "Rather than security information being converted from the format of one platform to the format of another, security information is passed between applications in the form of a token with a string data type. ...Making the token a string makes it platform and technology independent because the token has no header and therefore no application-specific header configuration." (Underlining added for emphasis.)

In other words, as shown above, Beck teaches the use of temporary user ID tokens associated with HTTP headers or cookie headers. Consequently, as discussed directly above, Beck does not teach or suggest "wherein the token contains user credentials encoded as a platform and application independent string data type," as recited in claim 1.

O'Donnell teaches the use of a user validation token. However, this user validation token is not the token recited in claim 1. Specifically, O'Donnell discloses in paragraph [0063], lines 1-8, "Referring to FIG. 2C, a validation token is also issued 240 in association with the proxy account. The validation token can be any unique identifier corresponding to the created proxy account. In one embodiment, the validation token comprises a random code with an appended identifier particular to the created proxy account. There are numerous alternatives for the validation token, including those that are merely the proxy account identifier." (Underlining added for emphasis.)

O'Donnell also discloses in paragraph [0071], lines 1-5, "The access site receives and verifies appropriate credentials corresponding to the proxy account, and then sends 328 a user validation token to the authorized user. Like the account validation token, the user validation token can be any kind of unique code, number or the like."

As shown above, O'Donnell teaches the use of a validation token with an appended identifier particular to the created proxy account. Consequently, this validation token is not a token containing "user credentials encoded as a platform and application independent string data type," as recited in claim 1.

Bhatia teaches the use of a Single Sign On (SSO) token that provides a listener mechanism for applications that need notification when the SSO token expires. Bhatia does not disclose, teach or suggest "wherein the token contains user credentials encoded as a platform and application independent string data type," as recited in claim 1. Therefore, for at least the above-described reasons, claim 1 is not unpatentable over Upton, Beck, O'Donnell and Bhatia and should be allowed.

II. Upton, Beck, O'Donnell and Bhatia do not disclose, teach or suggest a security application program interface and an application program interface coupled to a client application on a first operating system, and a server application on a second operating system to receive the token from the application program interface, the server application communicating with the authentication authority to validate the token.

Claim 1 (as previously presented) recites, in part, "a security application program interface and an application program interface coupled to a client application on a first operating system, ...and a server application on a second operating system to receive the token from the application program interface, the server application communicating with the authentication authority to validate the token..."

With respect to these features, the Office Action states in pertinent part:

> Upton discloses a system to provide application-to-application enterprise security, the system comprising:
>
> a security application program interface (Fig 4, Fig 5; Par 051, 0061, 0063, 0069; container; application security services; security

> provider interfaces) and an application program interface (fig 4, Fig 5; Par 0061, 0074, 0127; application interfaces) coupled to a client application operable on a first operating system, the security application program interface operable to provide a security credential (fig 4; Par 0061-0074, 0127-0130; container managed credentials; client application/ interfaces for storing, and providing security credentials);
>
> .
>
> .
>
> .
>
> a server application operable on a second operating system to receive the token from the application program interface, the server application communicating with the authentication authority to validate the token to enable the client application to user services of the server application (Par 0063-0065, 0104, 0114, 0130; JAAS, or SPI, or 3rd party validating/authenticating credentials).

Office Action dated September 18, 2008, Pages 3-4.

Contrary to the assertions made in the Office Action, Upton does not disclose, teach or suggest "a security application program interface and an application program interface coupled to a client application on a first operating system, ...and a server application on a second operating system to receive the token from the application program interface, the server application communicating with the authentication authority to validate the token..." as recited in (previously presented) claim 1. Specifically, the Office Action relied on the following disclosure in Upton to read on this element of claim 1:

For example, Upton, discloses in paragraph [0127], lines 1-7 and 18-30:

> FIG. 3 shows an example of a security architecture that can be used with systems and methods in accordance with embodiments of the invention. As shown therein, clients 302, 304 (which may be either physical hardware clients or software applications) may attempt to access a secured service or resource 306, such as a persistent directory server, via a transaction or application server 308. ...In any

> case, the connection attempt is received by the transaction server, often via an initial connection filter 320, and is passed to the security service 322. In accordance with the invention, the security service 322 is the focal point for security determination, including client and user level resource access, authorization, certification, privilege assessment and entitlement determination. Enterprise Java Beans (EJB's) 324, Web applications (WebApp's) 326, and other forms of applications may all use the security service through the use of containers. The security service handles calls from these containers to the protected resource, which in the case of FIG. 2 [sic].

Upton also discloses in paragraph [0128], lines 1-10 and 12-18:

> FIG. 4 illustrates an embodiment of the security service architecture 400 in greater detail. The security service augments the basic security services and features provided by the standard Java2 Enterprise Edition security set. As shown in this example, the basic java security set 402 includes security provider interfaces [SPI's] 404 for key storage, authentication, certificate validation, and secure sockets, among others. Customer applications 406 may be written to directly take advantage of the Java security layer and these SPI's. ...In accordance with the invention, customer applications are deployed in containers, for example, an EJB container 408 of a WebApp container 410. The containers communicate directly with the security service 414 (herein the same as security service 322), which in turn communicates with the Java security layer 402 and its security SPI's 404.

As shown above and in Figures 3 and 4, Upton teaches a client 302,304 that communicates with a transaction server 308 to access a secured resource 306. In particular, Upton discloses that the client 302,304 may attempt to access the secured resource 306 through the use of a security service, and a Java security set including security provider interfaces (SPI's) on the transaction server 308. Therefore, Upton may disclose performing a secure transaction with a secure resource 306 with a client 302,304, a transaction server 308, and the secure resource 306, each of which may be implemented on a separate physical computer. While each of these separate physical computers may have their own operating systems installed, Upton does not teach or suggest passing security credentials in a token between the separate computers where the operating systems on each of the separate computers are different (e.g., a heterogeneous environment with a Microsoft Windows operating system on one computer and a Unix operating system on another). Consequently, Upton does not disclose, teach or suggest "a security application program interface and an application program interface coupled to a client application on a first operating system, ...and a server application on a second operating system to receive the token from the application program interface, the server application communicating with the authentication authority to validate the token," as recited in claim 1.

Additionally, none of the secondary art cited, Beck, O'Donnell and Bhatia, discloses, teaches or suggests "a security application program interface and an application program interface coupled to a client application on a first operating system, ...and a server application on a second operating system to receive the token from the application program interface, the server application communicating with the authentication authority to validate the token," as recited in claim 1.

For at least the reasons established above in sections I and II, Applicants respectfully submit that independent claim 1 is not disclosed, taught or suggested by Upton in view O'Donnell or Bhatia. Accordingly, Applicants respectfully submit that claim 1 is patentable over Upton, O'Donnell and Bhatia and respectfully request allowance of this claim.

Claims depending from Claim 1:

Claims 2-3 were rejected under 35 USC § 103(a) as being unpatentable over Upton in view of O'Donnell.

Claim 8 was rejected under 35 USC § 103(a) as being unpatentable over Upton in view of O'Donnell further in view of Laferriere et al., U.S. Patent Application Publication No. 2005/0188212 (hereinafter, "Laferriere").

Claims 2-7 were rejected under 35 USC § 103(a) as being unpatentable over Upton in view of Bhatia.

Dependent claims 2-8 depend directly or indirectly from independent claim 1 and incorporate all of the limitations thereof. Accordingly, for at least the reasons established in sections I and II above, Applicants respectfully submit that claims 2-8 are not taught or suggested by Upton in view of O'Donnell. Laferriere and Bhatia, alone or in combination, do not cure the deficiencies of Upton and O'Donnell. Therefore, Applicants respectfully request allowance of these claims.

Claim 9:

Claim 9 was rejected under 35 USC § 103(a) as being unpatentable over Upton in view of O'Donnell.

Claim 9 was also rejected under 35 USC § 103(a) as being unpatentable over Upton in view of in view of Bhatia.

Claim 9 includes limitations substantially similar to the limitations discussed in sections I and II above. For example, claim 9 recites "the token contains user credentials encoded as a platform and application independent string data type." Claim 9 also recites "coupling a security application program interface and an application program interface to a client application on a first operating system ... providing, by the application program interface coupled to the client application on the first operating system, the token to a server application, the server application on a second operating system." Accordingly, the arguments of sections I and II are hereby repeated for claim 9.

For at least the reasons established above in sections I and II, Applicants respectfully submit that independent claim 9 is not taught or suggested by Upton in view O'Donnell and Bhatia does not cure the deficiencies of Upton and O'Donnell.

Claims Depending from Claim 9:

Dependent claims 11-12 and 24 were rejected under 35 USC § 103(a) as being unpatentable over Upton in view of O'Donnell.

Dependent claim 15 was rejected under 35 USC § 103(a) as being unpatentable over Upton in view of O'Donnell further in view of Laferriere.

Dependent claims 26 and 27 were rejected under 35 USC 103(a) as being unpatentable over Upton in view of O'Donnell further in view of Favazza et al., U.S. Patent Application Publication No. 2004/0139319 (hereinafter "Favazza").

Dependent claims 13-14, and 16-25 were rejected under 35 USC 103(a) as being unpatentable over Upton in view of Bhatia.

Dependent claims 11-27 depend directly or indirectly from independent claim 9 and incorporate all of the limitations thereof. Accordingly, for at least the reasons established in sections I and II above, Applicants respectfully submit that claims 11-27 are not taught or suggested by Upton in view O'Donnell, and Laferriere, Favazza or Bhatia, alone or in combination, do not cure the deficiencies of Upton and O'Donnell. Therefore, Applicants respectfully request allowance of these claims.

Claim 28:

Claim 28 was rejected under 35 USC § 103(a) as being unpatentable over Upton in view of O'Donnell.

Claim 28 was also rejected under 35 USC § 103(a) as being unpatentable over Upton in view of Bhatia.

Claim 28 includes limitations substantially similar to the limitations discussed in sections I and II above. For example, claim 28 recites "wherein the token contains user credentials encoded as a platform and application independent string data type." Claim 28 also recites "a first security application program interface coupled to the first application on the first operating system ... a second application program interface coupled to a second application on a second operating system; a second security application program interface coupled to the second application on the second operating system, to provide a second security credential." Accordingly, the arguments of sections I and II are hereby repeated for claim 28.

For at least the reasons established above in sections I and II, Applicants respectfully submit that independent claim 28 is not taught or suggested by Upton in O'Donnell. Bhatia does not cure the deficiencies of Upton and O'Donnell. Therefore, Applicants respectfully request allowance of this claim.

Claims Depending from Claim 28:

Claim 29 was rejected under 35 USC § 103(a) as being unpatentable over Upton in view of O'Donnell.

Claims 29-33 were also rejected under 35 USC § 103(a) as being unpatentable over Upton in view of Bhatia.

Dependent claims 29-33 depend directly or indirectly from independent claim 28 and incorporate all of the limitations thereof. Accordingly, for at least the reasons established in sections I and II above, Applicants respectfully submit that claims 29-33 are not taught or suggested by Upton in view of O'Donnell. Bhatia does not cure the deficiencies of Upton and O'Donnell. Therefore, Applicants respectfully request allowance of these claims.

Response to Rejections under Section 102

Claim 1:

Claim 1 was rejected under 35 USC § 102(e) as being anticipated by Bhatia et al., U.S. Patent No. 7,249,375 (hereinafter "Bhatia").

III. Bhatia does not disclose, teach or suggest wherein the token contains user credentials encoded as a platform and application independent string data type.

Claim 1 (as previously presented) of the pending application recites in part "an authentication authority receiving the security credential from the security application program interface, the authentication authority further generates a token ... wherein the token contains user credentials encoded as a platform and application independent string data type."

With respect to claim 1, the Office Action states, in pertinent part:

> an authentication authority receiving the security credential from the security application program interface, the authentication authority (col 3, starts at line 6, SSO servers) further generates a token and communicates the token to the security application program interface where the security credential is valid, wherein the token contains user credential encoded as a platform and application independent string data type (Fig 3; Col 3, starts at line 16; XML/ security token for authentication);

Office Action dated September 18, 2008, Page 16.

Contrary to the assertions made in the Office Action, Bhatia does not disclose, teach or suggest "an authentication authority receiving the security credential from the security application program interface, the authentication authority further generates a token ... wherein the token contains user credentials encoded as a platform and application independent string data type." as recited in (previously presented) claim 1. Specifically, the Office Action relied on the column 3 in Bhatia to read on this element of claim 1 and referred to the SSO and XMO/security token (which is actually disclosed in column 4). However, a close reading of the entirety of Bhatia reveals that it does not disclose "the token contains user credential encoded as a platform and application independent string data type," but a token in multiple formats. Bhatia discloses in column 3, line 33-34, that the "SSO server **106** can issue tokens in multiple formats based upon the capabilities of the target system." The only reason to issue tokens in multiple formats is if the tokens are not "platform and application independent string data type" but are specific to certain platforms and/or applications.

For at least the reasons established above in section III, Applicants respectfully submit that independent claim 1 is not anticipated Bhatia. Therefore, Applicants respectfully request allowance of this claim.

Claim 9:

Claim 9 was rejected under 35 USC § 102(e) as being anticipated by Bhatia.

Claim 9 includes limitations substantially similar to the limitations discussed in sections III above. For example, claim 9 recites "wherein the token contains user credentials encoded as a platform and application independent string data type." Accordingly, the arguments of section III are hereby repeated for claim 9.

For at least the reasons established above in section III, Applicants respectfully submit that independent claim 9 is not anticipated Bhatia. Therefore, Applicants respectfully request allowance of this claim.

Claim 28:

Claim 28 was rejected under 35 USC § 102(e) as being anticipated by Bhatia.

Claim 28 includes limitations substantially similar to the limitations discussed in section III above. For example, claim 28 recites wherein the token contains user credentials encoded as a platform and application independent string data type." Accordingly, the arguments of section III are hereby repeated for claim 28.

For at least the reasons established above in section III, Applicants respectfully submit that independent claim 28 is not anticipated Bhatia. Therefore, Applicants respectfully request allowance of this claim.

Conclusion

Applicants respectfully submit that the pending application is in condition for allowance for the reasons stated above. If the Examiner has any questions or comments or otherwise feels it would be helpful in expediting the application, the Examiner is encouraged to telephone the undersigned at (972) 731-2288.

The Commissioner is hereby authorized to charge payment of any further fees associated with any of the foregoing papers submitted herewith, or to credit any overpayment thereof, to Deposit Account No. 21-0765, Sprint.

Respectfully submitted,

Date: December 18, 2008

/Michael W. Piper/
Michael W. Piper
Reg. No. 39,800

ATTORNEY FOR APPLICANTS

CONLEY ROSE, P.C.
5601 Granite Parkway, Suite 750
Plano, Texas 75024
(972) 731-2288
(972) 731-2289 (facsimile)